

Original lasso Slides

Lasso Regression: Regularization for feature selection

CS229: Machine Learning
Carlos Guestrin
Stanford University

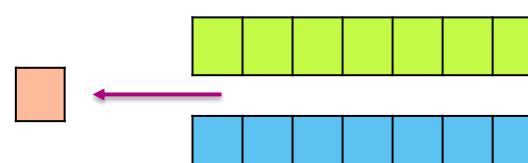
Slides include content developed by and co-developed with
Emily Fox

Feature selection task

Why might you want to perform feature selection?

Efficiency:

- If size(w) = 100B, each prediction is expensive
- If \hat{w} sparse , computation only depends on # of non-zeros

$$\hat{y}_i = \sum_{\hat{w}_j \neq 0} \hat{w}_j h_j(x_i)$$


Interpretability:

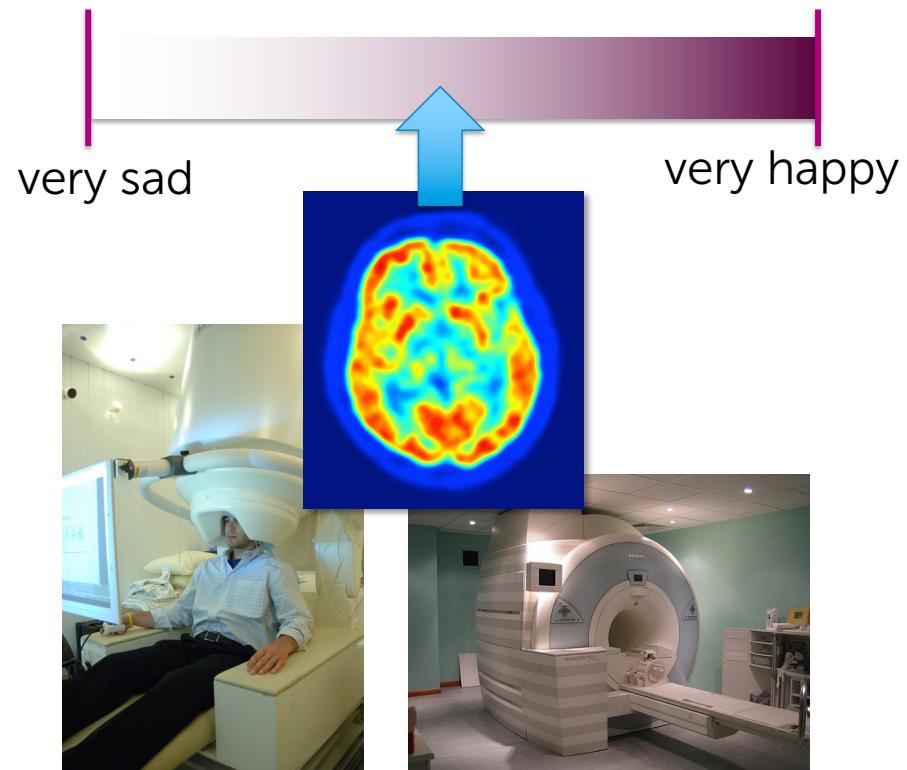
- Which features are relevant for prediction?

Sparsity: Housing application



Lot size	Dishwasher
Single Family	Garbage disposal
Year built	Microwave
Last sold price	Range / Oven
Last sale price/sqft	Refrigerator
Finished sqft	Washer
Unfinished sqft	Dryer
Finished basement sqft	Laundry location
# floors	Heating type
Flooring types	Jetted Tub
Parking type	Deck
Parking amount	Fenced Yard
Cooling	Lawn
Heating	Garden
Exterior materials	Sprinkler System
Roof type	:
Structure style	

Sparsity: Reading your mind

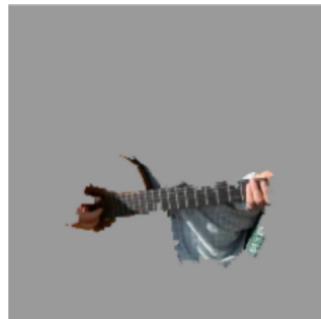


Activity in which brain regions can predict happiness?

Explaining Predictions



$P(\text{Electric Guitar}) = 0.32$



$P(\text{Acoustic Guitar}) = 0.24$



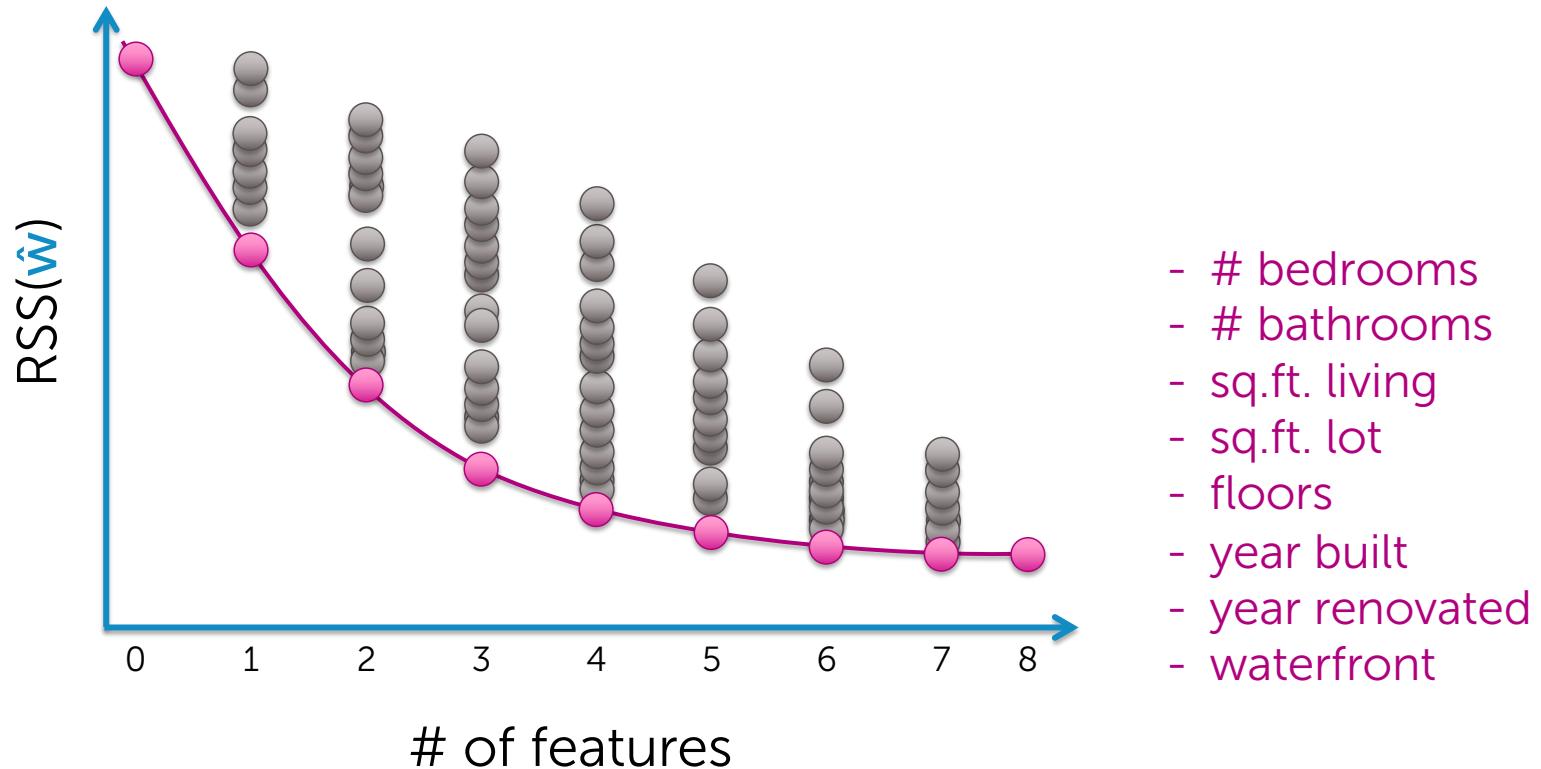
$P(\text{Dog}) = 0.21$



“Why should I trust you?”: Explaining the Predictions of Any Classifier. Ribeiro, Singh & G. KDD 16

Option 1: All subsets or greedy variants

Find best model of for each size



Complexity of “all subsets”

$$y_i = \varepsilon_i$$

$$y_i = w_0 h_0(x_i) + \varepsilon_i$$

$$y_i = w_1 h_1(x_i) + \varepsilon_i$$

:

$$y_i = w_0 h_0(x_i) + w_1 h_1(x_i) + \varepsilon_i$$

:

$$y_i = w_0 h_0(x_i) + w_1 h_1(x_i) + \dots + w_D h_D(x_i) + \varepsilon_i$$

[0 0 0 ... 0 0 0]

[1 0 0 ... 0 0 0]

[0 1 0 ... 0 0 0]

:

[1 1 0 ... 0 0 0]

:

[1 1 1 ... 1 1 1]

}

$2^8 = 256$
 $2^{30} = 1,073,741,824$
 $2^{1000} = 1.071509 \times 10^{301}$
 $2^{100B} = \text{HUGE!!!!!!}$

2^D

Typically,
computationally
infeasible

Greedy algorithms

Forward stepwise:

Starting from simple model and iteratively add features most useful to fit

Backward stepwise:

Start with full model and iteratively remove features least useful to fit

Combining forward and backward steps:

In forward algorithm, insert steps to remove features no longer as important

Lots of other variants, too.

Option 2: Regularize

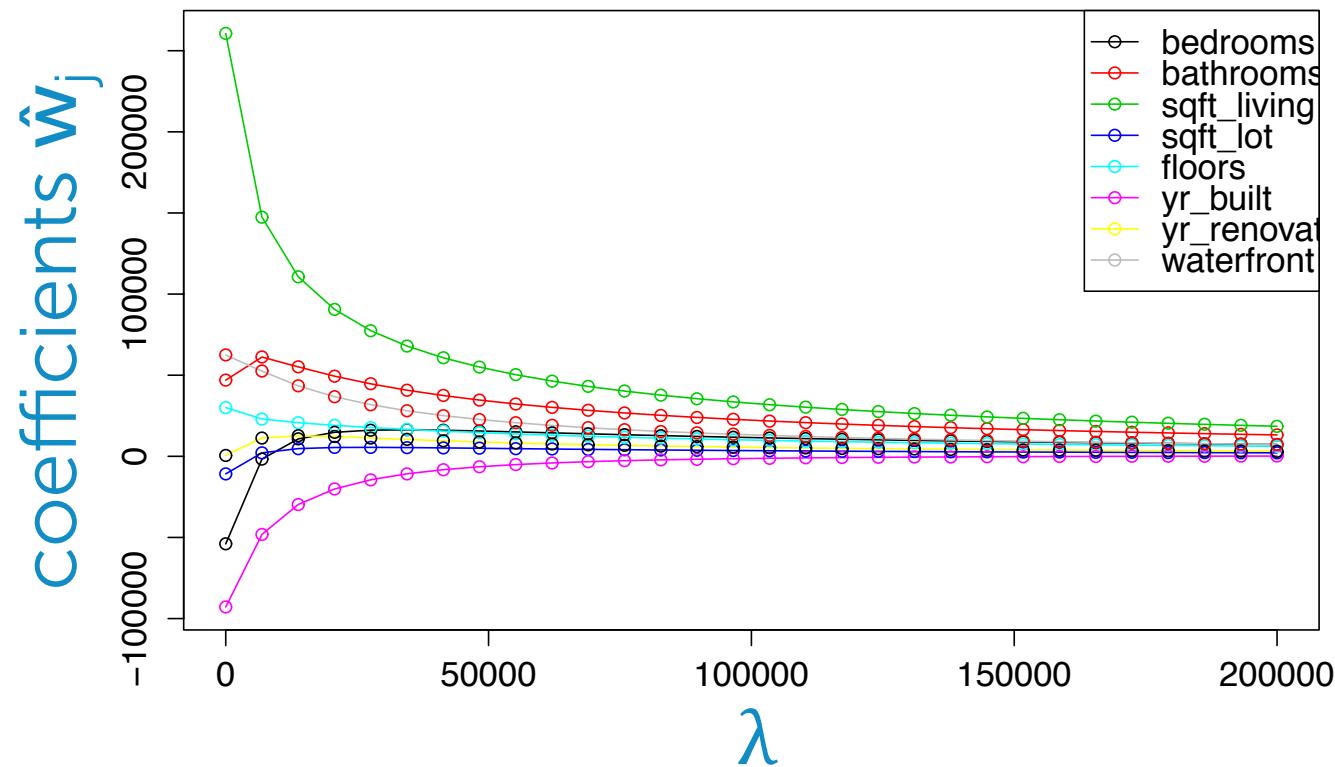
Ridge regression: L_2 regularized regression

Total cost =

$$\text{measure of fit} + \lambda \text{ measure of magnitude of coefficients}$$


RSS(w) $\|w\|_2^2 = w_0^2 + \dots + w_D^2$

Coefficient path – ridge



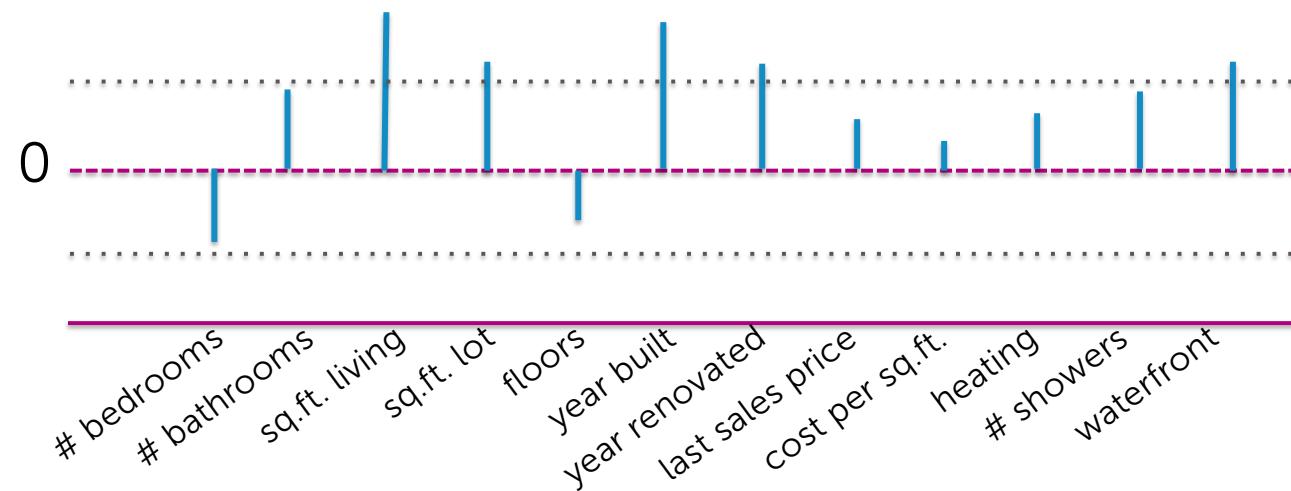
Using regularization for feature selection

Instead of searching over a discrete set of solutions, can we use **regularization**?

- Start with full model (all possible features)
- “Shrink” some coefficients *exactly to 0*
 - i.e., knock out certain features
- Non-zero coefficients indicate “selected” features

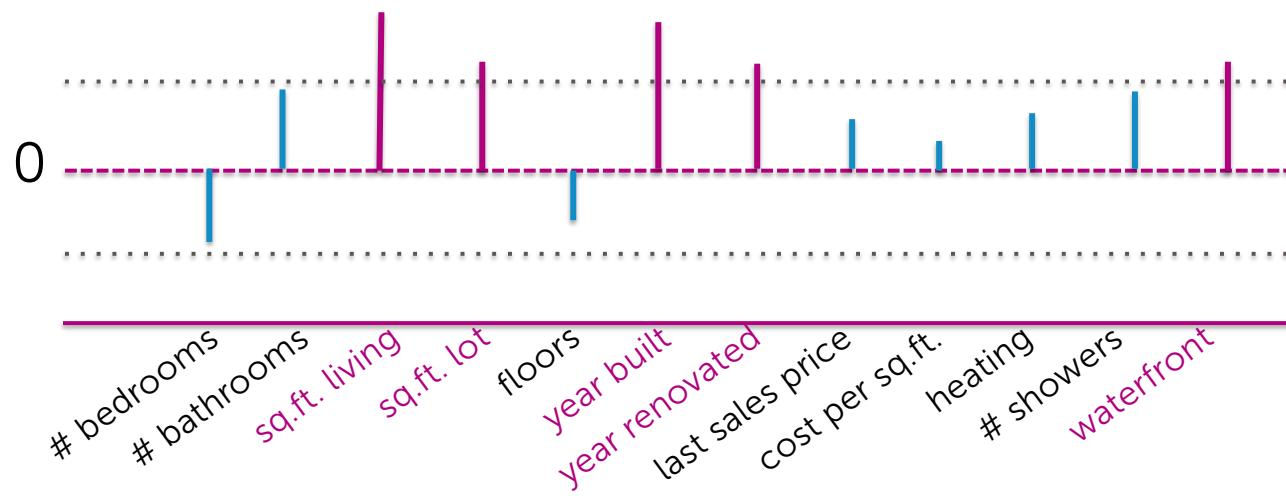
Thresholding ridge coefficients?

Why don't we just set small ridge coefficients to 0?



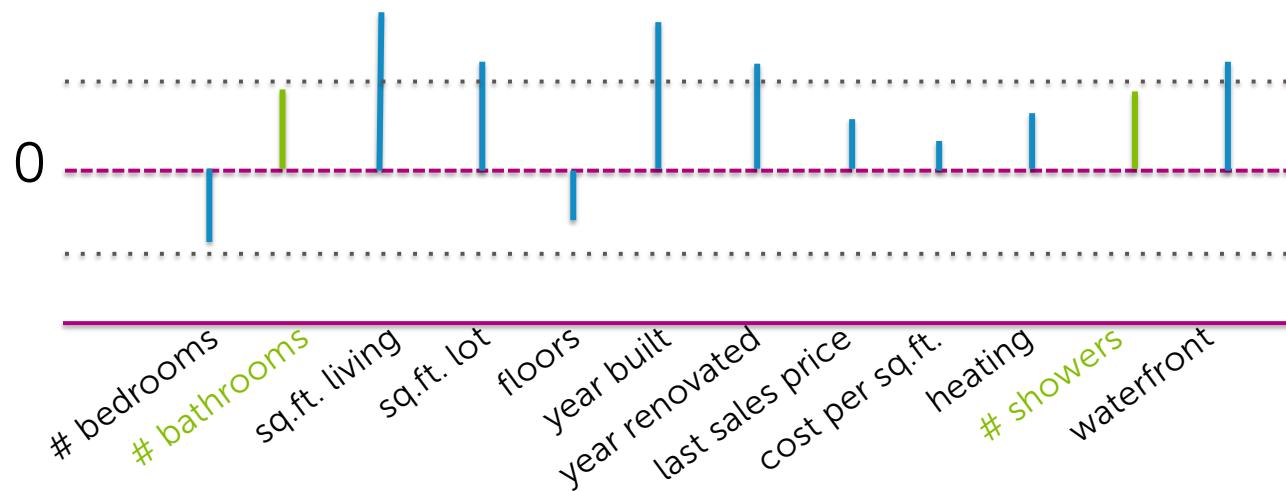
Thresholding ridge coefficients?

Selected features for a given threshold value



Thresholding ridge coefficients?

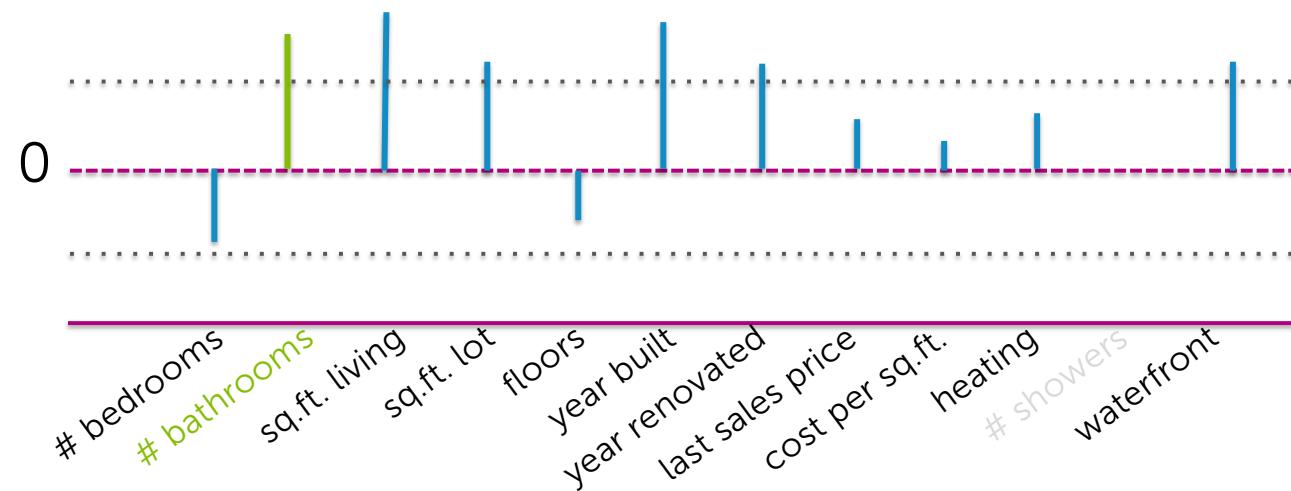
Let's look at two related features...



Nothing measuring bathrooms was included!

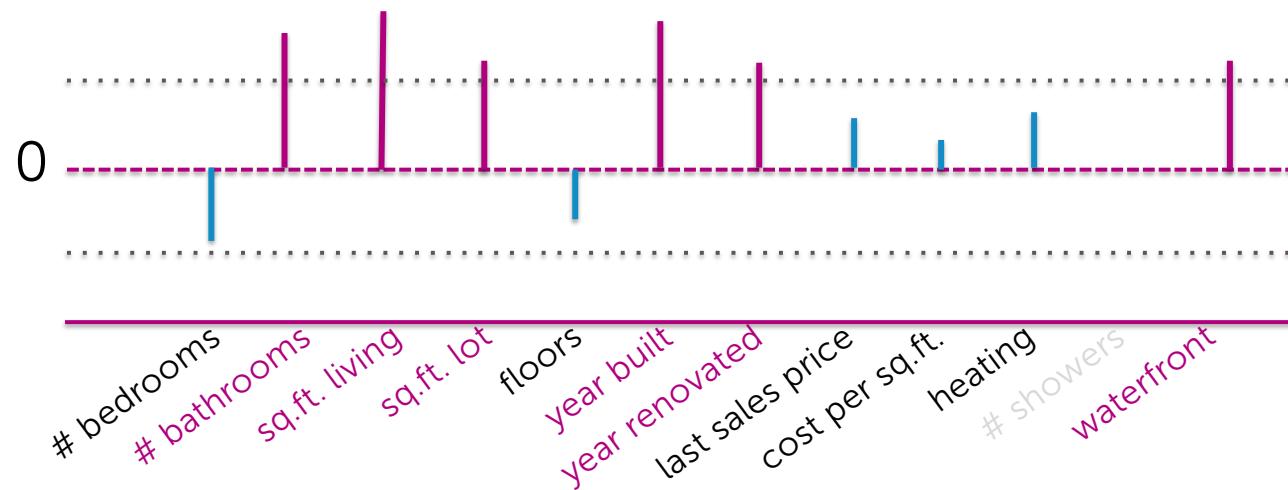
Thresholding ridge coefficients?

If only one of the features had been included...



Thresholding ridge coefficients?

Would have included bathrooms in selected model



Can regularization lead directly to sparsity?

Try this cost instead of ridge...

Total cost =

measure of fit + λ measure of magnitude of coefficients

RSS(w)

$\|w\|_1 = |w_0| + \dots + |w_D|$

Leads to **sparse** solutions!

Lasso regression
(a.k.a. L_1 regularized regression)

Lasso regression: L_1 regularized regression

Just like ridge regression, solution is governed by a continuous parameter λ

$$\text{RSS}(\mathbf{w}) + \lambda \|\mathbf{w}\|_1$$

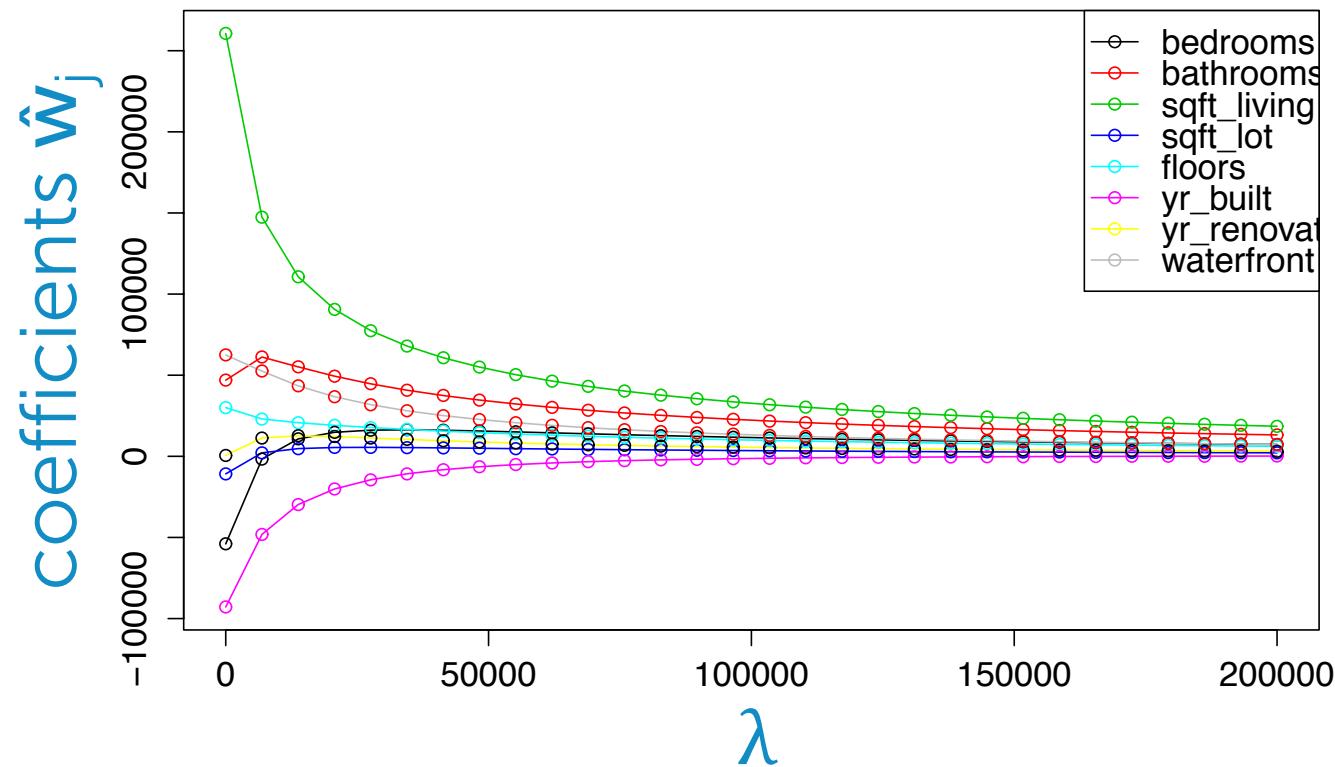
 tuning parameter = balance of fit and sparsity

If $\lambda=0$:

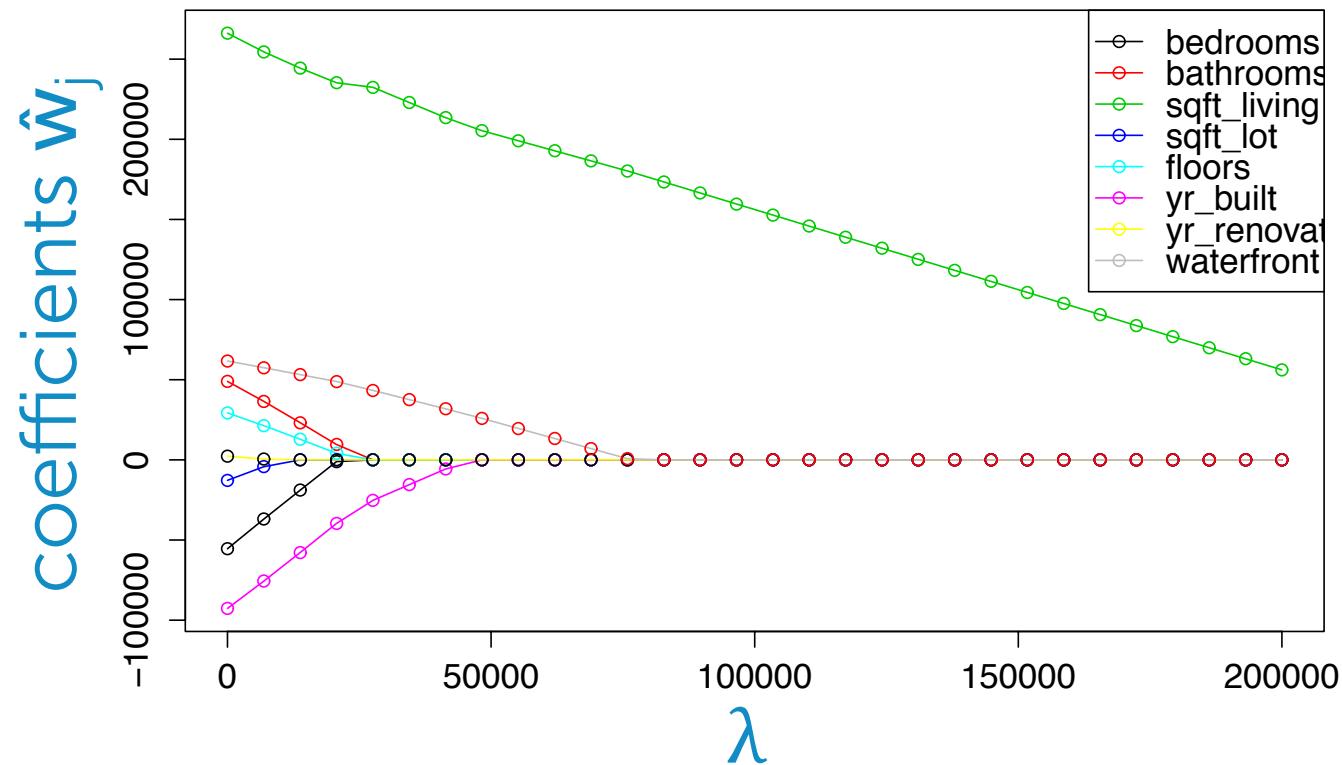
If $\lambda=\infty$:

If λ in between:

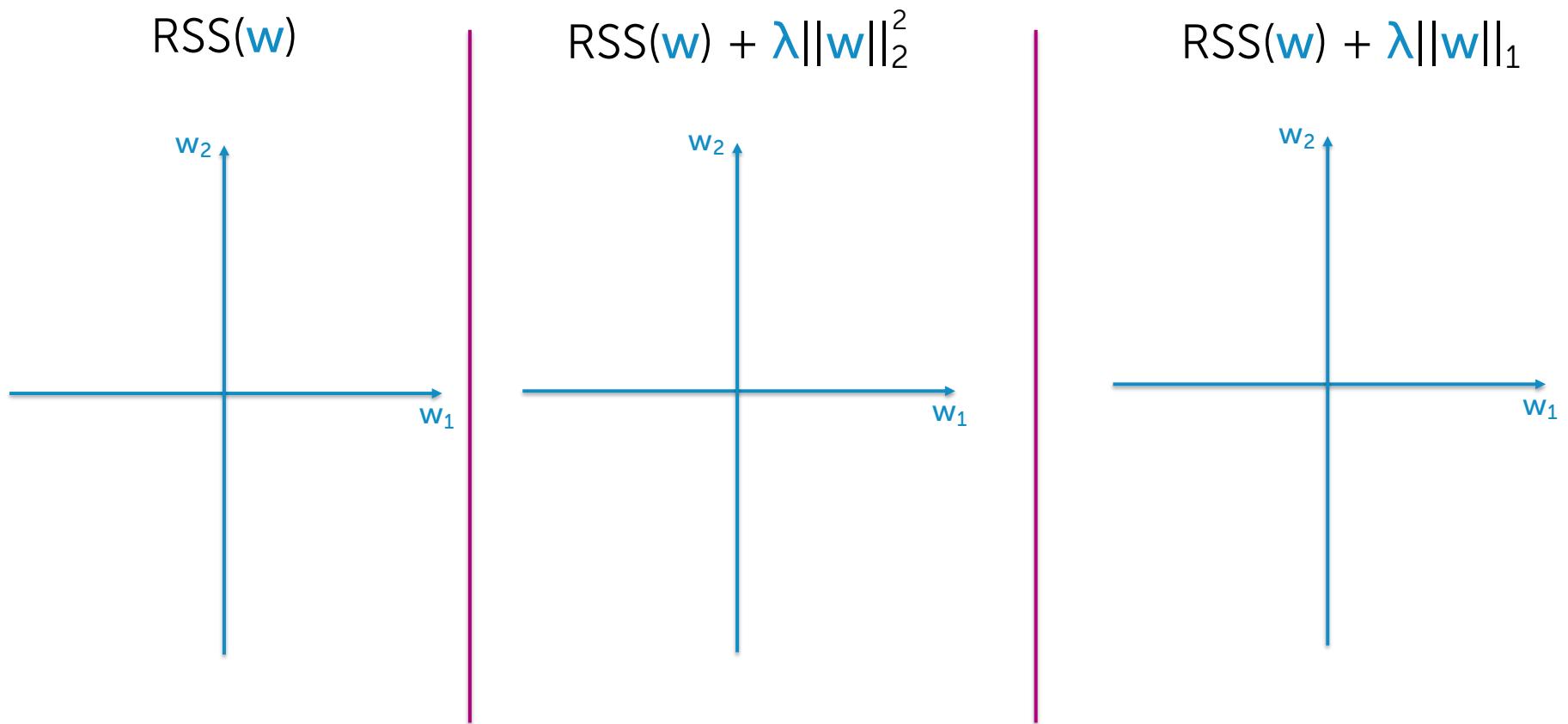
Coefficient path – ridge



Coefficient path – lasso



Intuitive difference between Lasso and Ridge



Practical concerns with lasso

Debiasing lasso

Lasso shrinks coefficients relative to LS solution

→ more bias, less variance

Can reduce bias as follows:

1. Run lasso to select features
2. Run least squares regression with only selected features

"Relevant" features no longer shrunk relative to LS fit of same reduced model

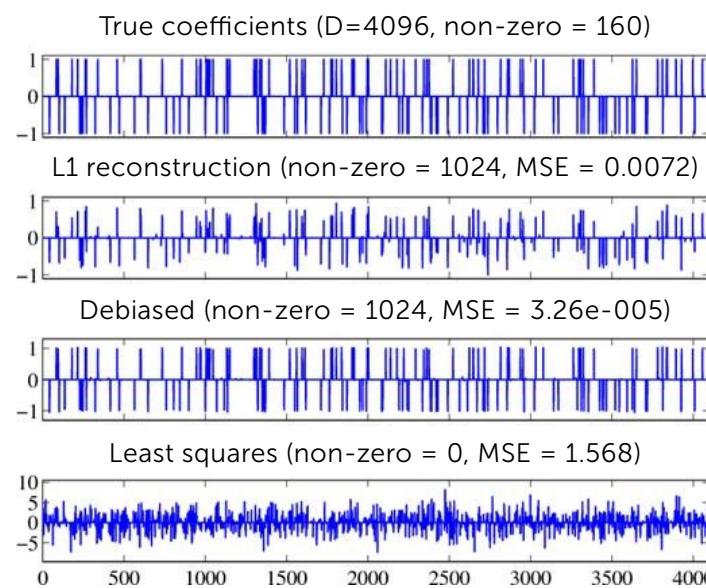


Figure used with permission of Mario Figueiredo
(captions modified to fit course)

Issues with standard lasso objective

1. With group of highly correlated features, lasso tends to select amongst them arbitrarily
 - Often prefer to select all together
2. Often, empirically ridge has better predictive performance than lasso, but lasso leads to sparser solution

Elastic net aims to address these issues

- hybrid between lasso and ridge regression
- uses L_1 and L_2 penalties

See [Zou & Hastie '05](#) for further discussion

Summary for feature selection and lasso regression

Impact of feature selection and lasso

Lasso has changed machine learning,
statistics, & electrical engineering

But, for feature selection in general, be **careful about interpreting selected features**

- selection only considers features included
- sensitive to correlations between features
- result depends on algorithm used
- there are theoretical guarantees for lasso under certain conditions

What you can do now...

- Describe “all subsets” and greedy variants for feature selection
- Analyze computational costs of these algorithms
- Formulate lasso objective
- Describe what happens to estimated lasso coefficients as tuning parameter λ is varied
- Interpret lasso coefficient path plot
- Contrast ridge and lasso regression
- Implement K-fold cross validation to select lasso tuning parameter λ

Annotated lasso Slides

Lasso Regression: Regularization for feature selection

CS229: Machine Learning
Carlos Guestrin
Stanford University

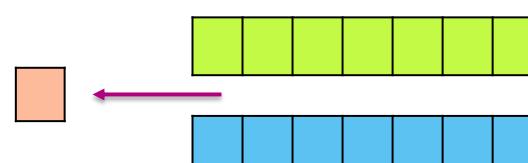
Slides include content developed by and co-developed with
Emily Fox

Feature selection task

Why might you want to perform feature selection?

Efficiency:

- If size(\mathbf{w}) = 100B, each prediction is expensive
- If $\hat{\mathbf{w}}$ sparse , computation only depends on # of non-zeros

$$\hat{y}_i = \sum_{\hat{w}_j \neq 0} \hat{w}_j h_j(x_i)$$


The diagram shows two horizontal arrays. The top array consists of seven light green squares. The bottom array consists of seven blue squares. A pink arrow points from the orange square to the bottom array, indicating that only the non-zero elements in the weight vector are used in the computation.

Interpretability:

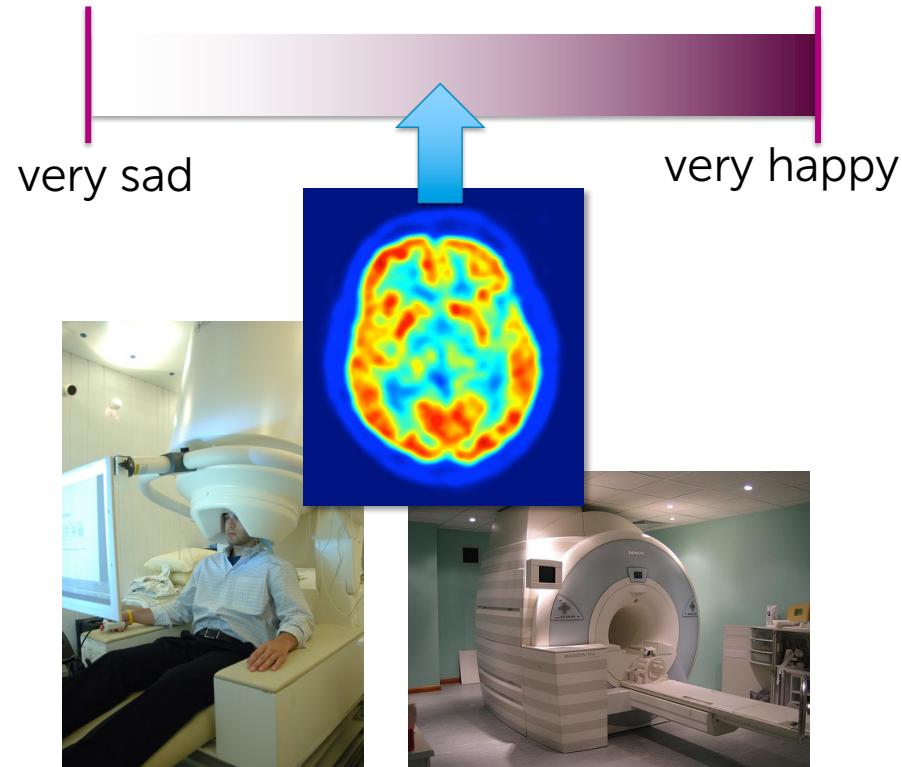
- Which features are relevant for prediction?

Sparsity: Housing application



Lot size	Dishwasher
Single Family	Garbage disposal
Year built	Microwave
Last sold price	Range / Oven
Last sale price/sqft	Refrigerator
Finished sqft	Washer
Unfinished sqft	Dryer
Finished basement sqft	Laundry location
# floors	Heating type
Flooring types	Jetted Tub
Parking type	Deck
Parking amount	Fenced Yard
Cooling	Lawn
Heating	Garden
Exterior materials	Sprinkler System
Roof type	:
Structure style	

Sparsity: Reading your mind

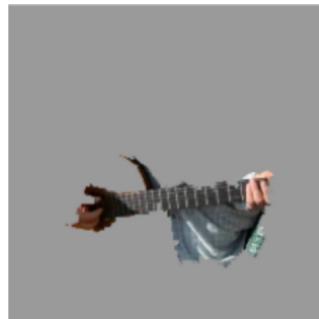


Activity in which brain
regions can predict
happiness?

Explaining Predictions



$P(\text{Electric Guitar}) = 0.32$



$P(\text{Acoustic Guitar}) = 0.24$



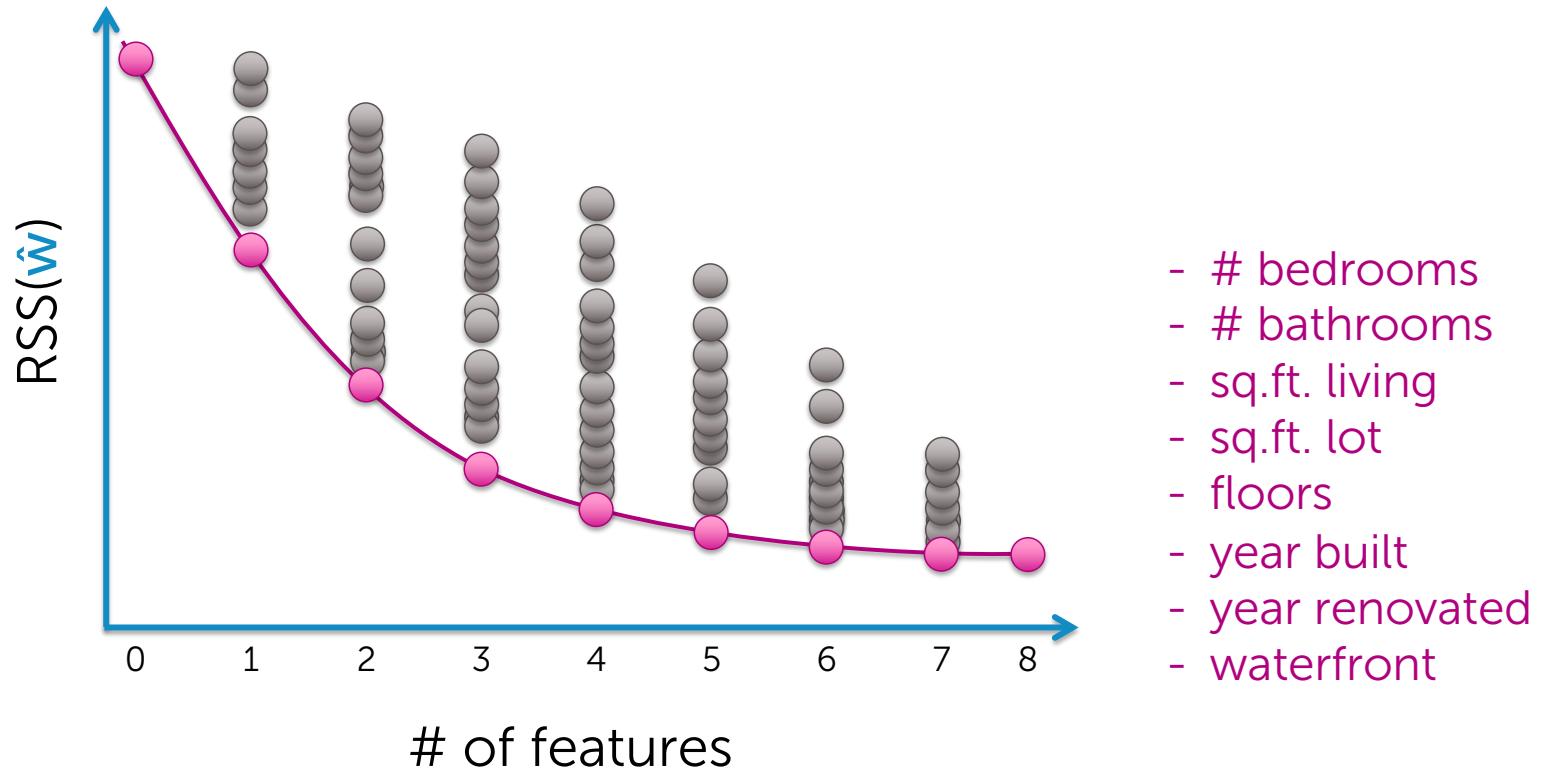
$P(\text{Dog}) = 0.21$



“Why should I trust you?”: Explaining the Predictions of Any Classifier. Ribeiro, Singh & G. KDD 16

Option 1: All subsets or greedy variants

Find best model of for each size



Complexity of “all subsets”

$$y_i = \varepsilon_i$$

$$y_i = w_0 h_0(x_i) + \varepsilon_i$$

$$y_i = w_1 h_1(x_i) + \varepsilon_i$$

:

$$y_i = w_0 h_0(x_i) + w_1 h_1(x_i) + \varepsilon_i$$

:

$$y_i = w_0 h_0(x_i) + w_1 h_1(x_i) + \dots + w_D h_D(x_i) + \varepsilon_i$$

[0 0 0 ... 0 0 0]

[1 0 0 ... 0 0 0]

[0 1 0 ... 0 0 0]

:

[1 1 0 ... 0 0 0]

:

[1 1 1 ... 1 1 1]

}

$2^8 = 256$
 $2^{30} = 1,073,741,824$
 $2^{1000} = 1.071509 \times 10^{301}$
 $2^{100B} = \text{HUGE!!!!!!}$

2^D

Typically,
computationally
infeasible

Greedy algorithms

Forward stepwise:

Starting from simple model and iteratively add features most useful to fit

Backward stepwise:

Start with full model and iteratively remove features least useful to fit

Combining forward and backward steps:

In forward algorithm, insert steps to remove features no longer as important

Lots of other variants, too.

Option 2: Regularize

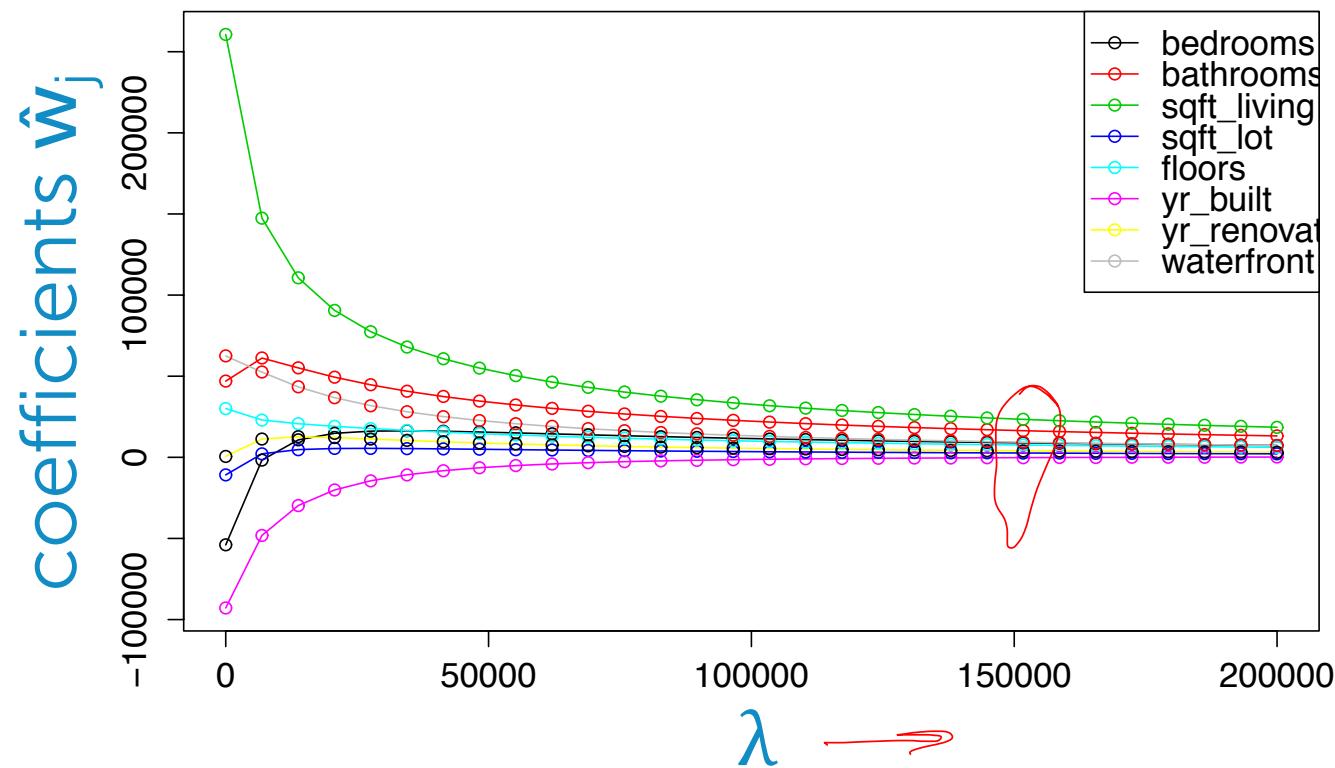
Ridge regression: L_2 regularized regression

Total cost =

$$\text{measure of fit} + \lambda \text{ measure of magnitude of coefficients}$$

The equation shows the total cost function for Ridge regression. It consists of two terms: a blue bracket underlined 'measure of fit' followed by a plus sign, and a red bracket underlined 'measure of magnitude of coefficients'. The first term is labeled $\text{RSS}(\mathbf{w})$. The second term is labeled $\|\mathbf{w}\|_2^2 = w_0^2 + \dots + w_D^2$. A red circle highlights the regularization parameter λ .

Coefficient path – ridge



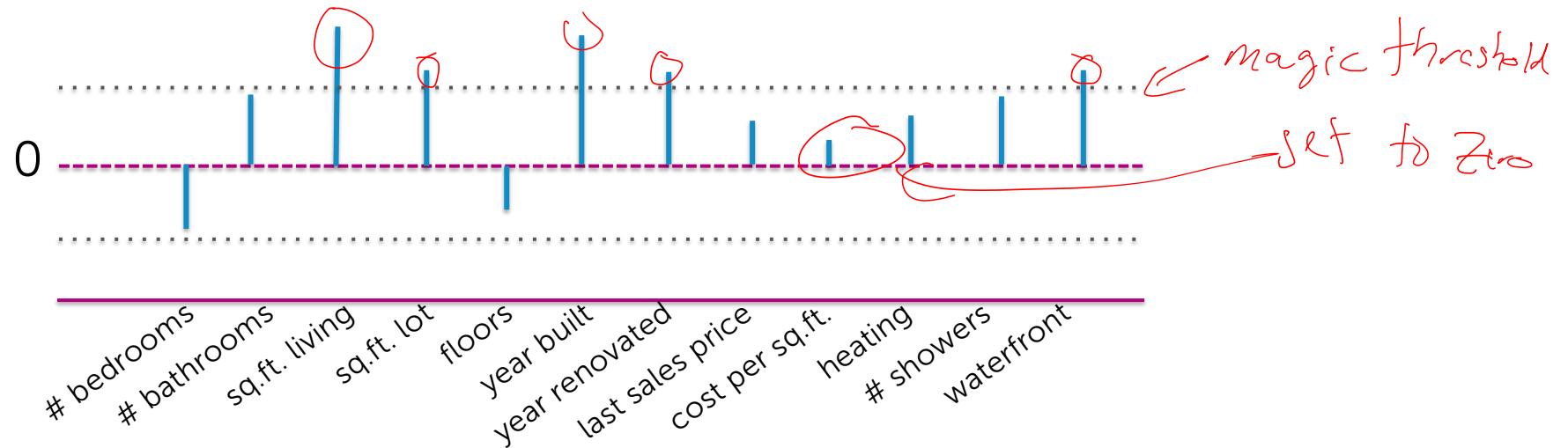
Using regularization for feature selection

Instead of searching over a **discrete** set of solutions, can we use **regularization**?

- Start with full model (all possible features)
- “Shrink” some coefficients *exactly to 0*
 - i.e., knock out certain features
- Non-zero coefficients indicate “selected” features

Thresholding ridge coefficients?

Why don't we just set small ridge coefficients to 0?



Thresholding ridge coefficients?

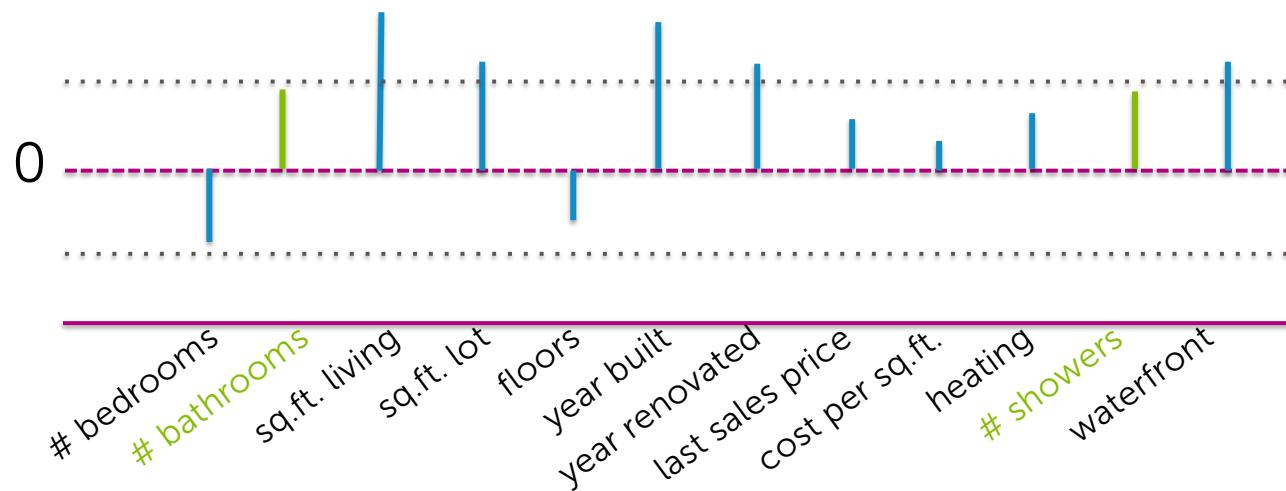
Selected features for a given threshold value



bathrooms
Seems
important

Thresholding ridge coefficients?

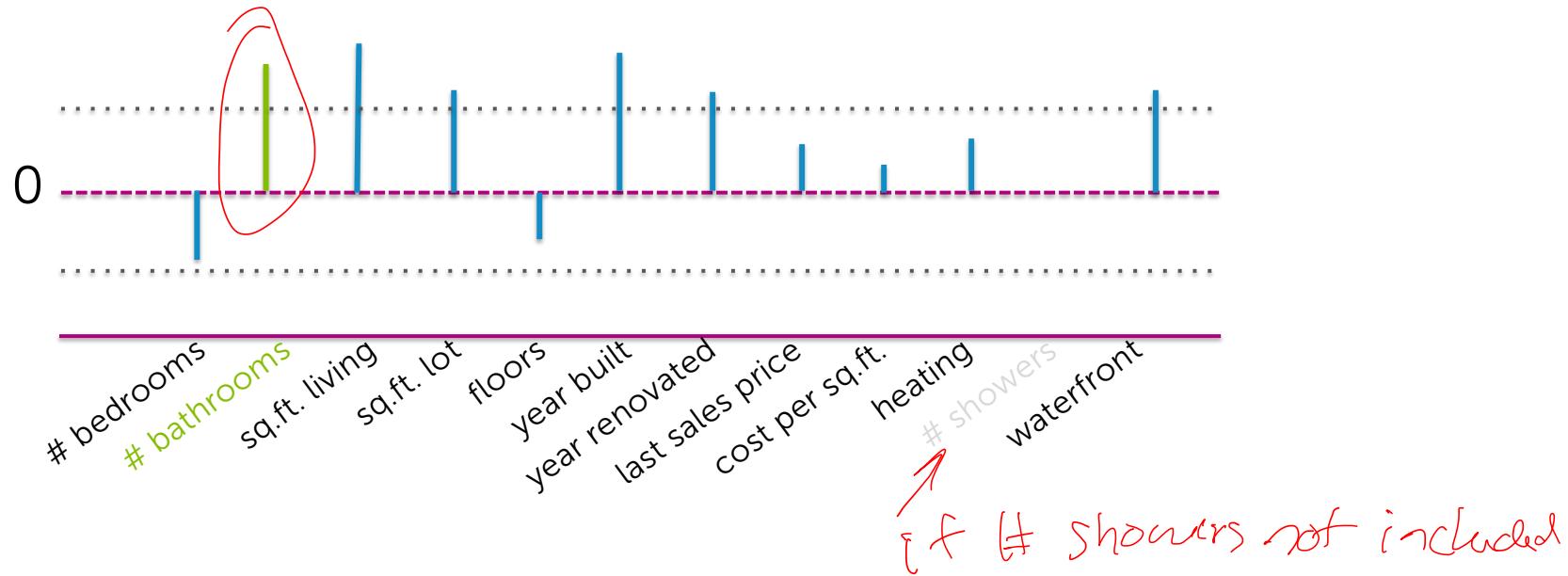
Let's look at two related features...



Nothing measuring bathrooms was included!

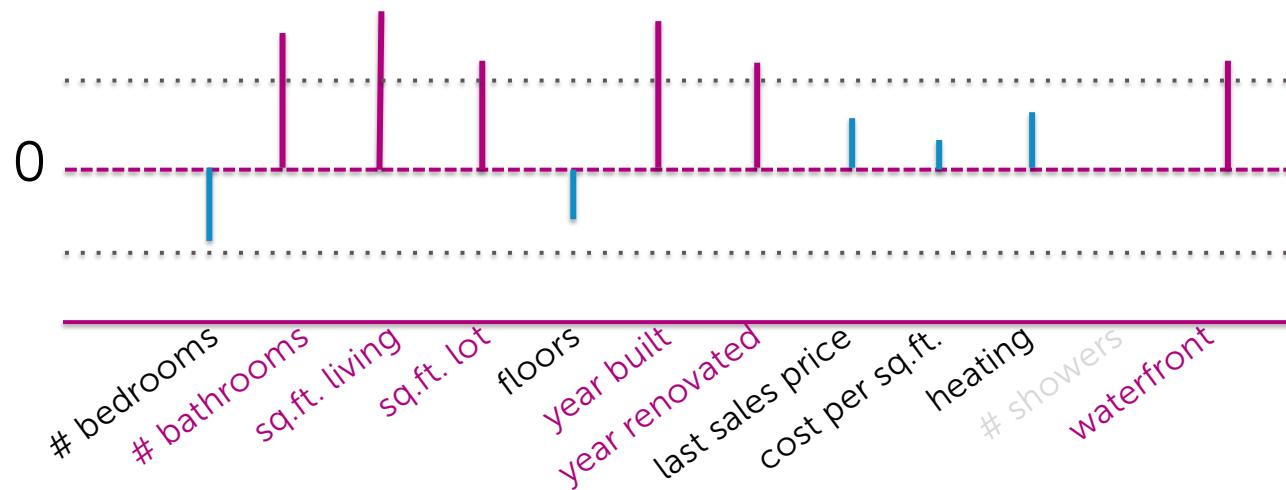
Thresholding ridge coefficients?

If only one of the features had been included...



Thresholding ridge coefficients?

Would have included bathrooms in selected model



Can regularization lead directly to sparsity?

Try this cost instead of ridge...

Total cost =

measure of fit + λ measure of magnitude of coefficients

RSS(w)

$\|w\|_1 = |w_0| + \dots + |w_D|$

Leads to **sparse** solutions!

Lasso regression
(a.k.a. L_1 regularized regression)

Lasso regression: L_1 regularized regression

Just like ridge regression, solution is governed by a continuous parameter λ

$$\text{RSS}(\mathbf{w}) + \lambda \|\mathbf{w}\|_1$$

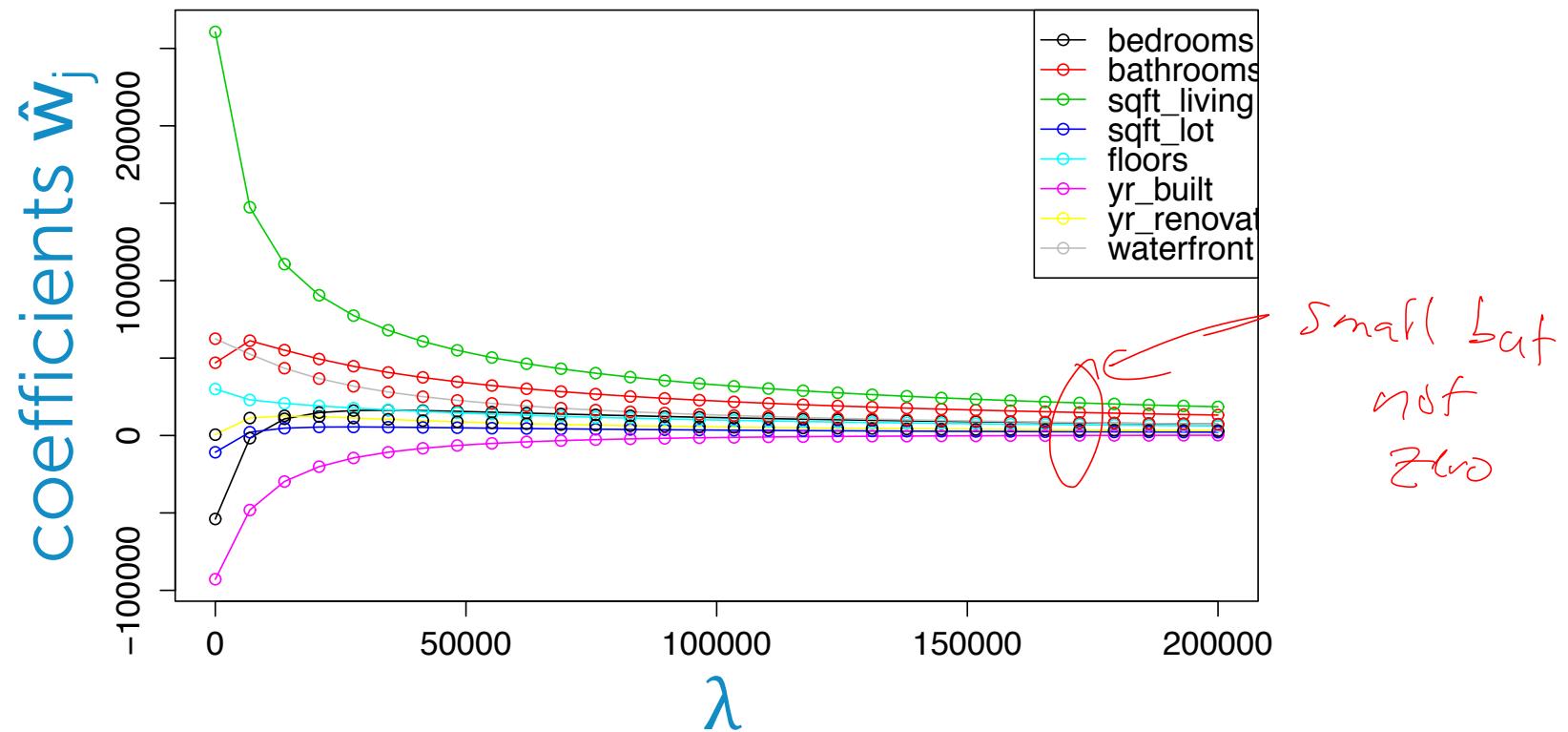
↑ tuning parameter = balance of fit and sparsity

If $\lambda=0$: $\hat{\mathbf{w}}_{\text{Lasso}} = \hat{\mathbf{w}}_{\text{RSS}}$

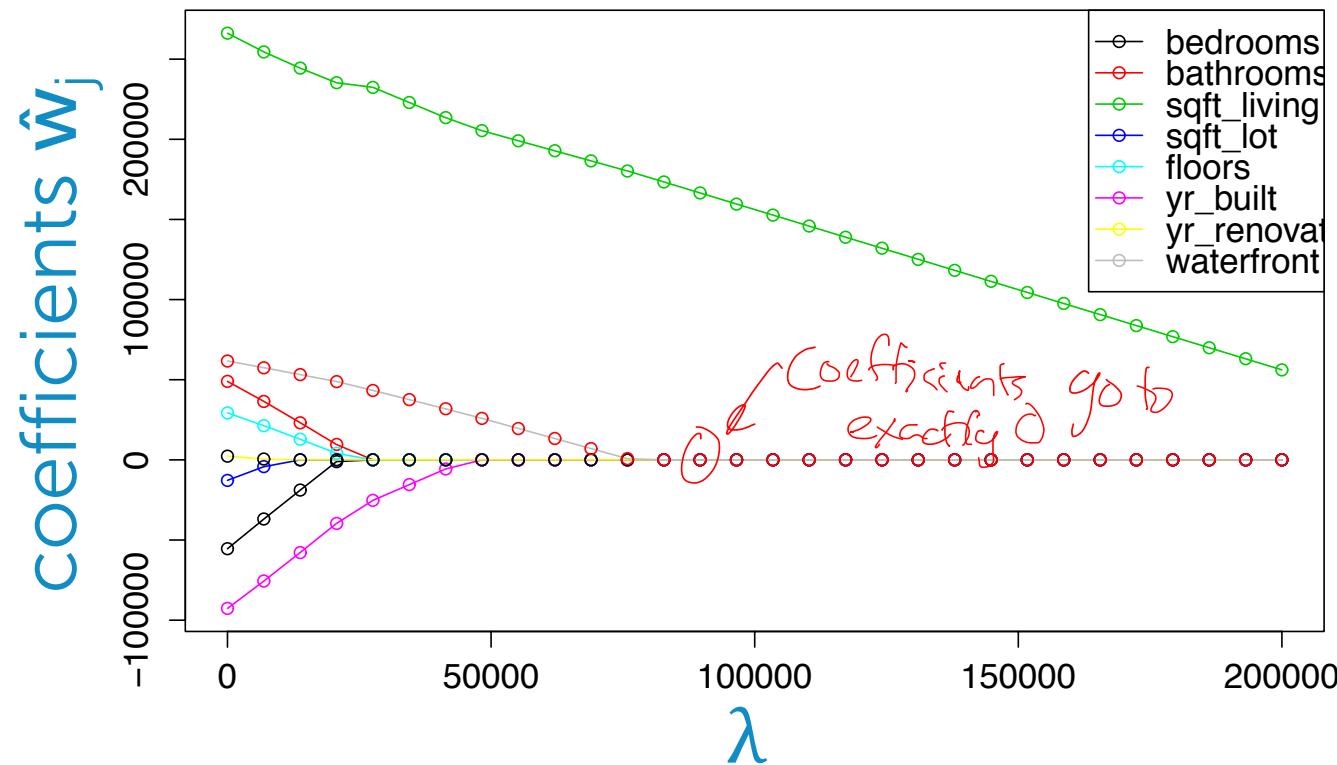
If $\lambda=\infty$: $\hat{\mathbf{w}}_{\text{Lasso}} = \mathbf{0}$

If λ in between: $\|\hat{\mathbf{w}}_{\text{Lasso}}\|_1 < \|\hat{\mathbf{w}}_{\text{RSS}}\|_1$

Coefficient path – ridge

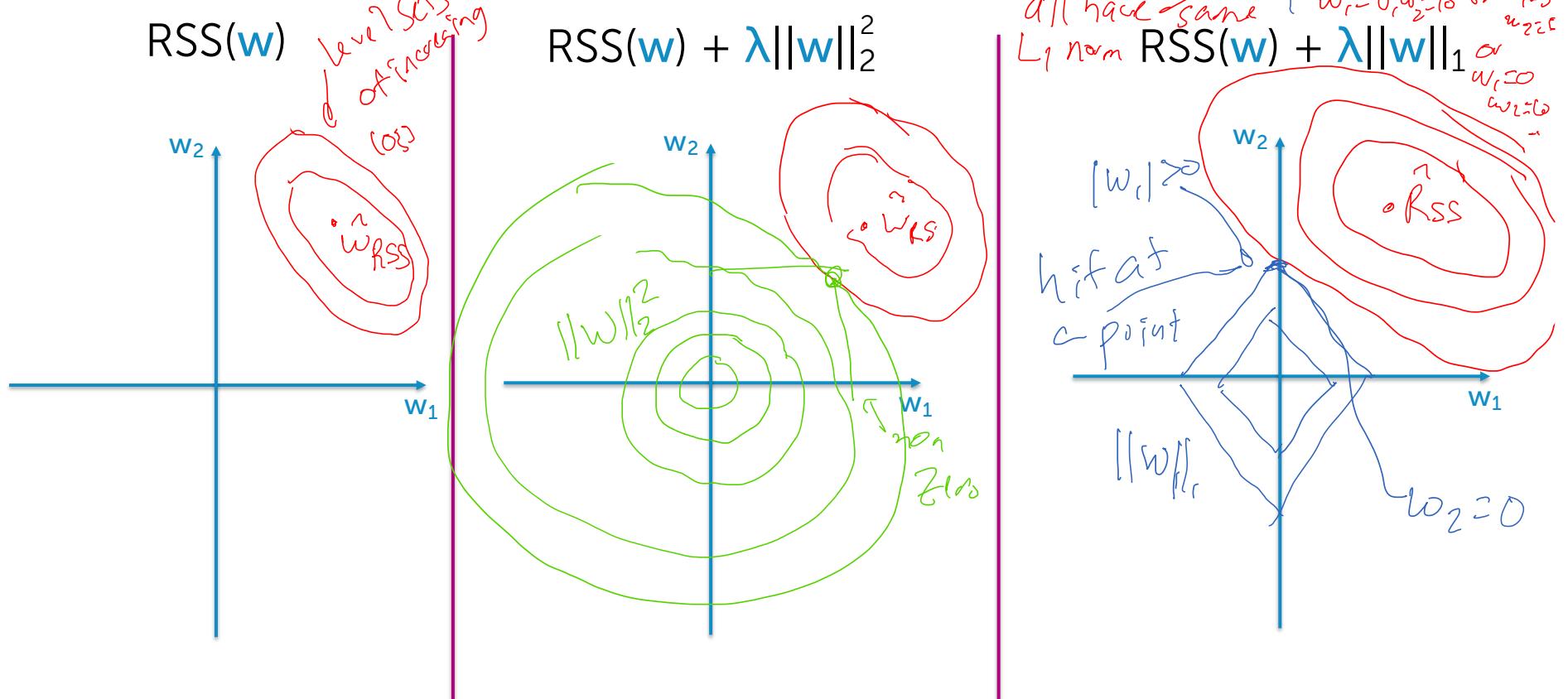


Coefficient path – lasso



what happens, say, with identical features $x^{(1)} = x^{(2)}$? // Solution is different:
 if weight is 0, both $w_1=0, w_2=0$ or $w_1=5, w_2=5$

Intuitive difference between Lasso and Ridge



Practical concerns with lasso

Debiasing lasso

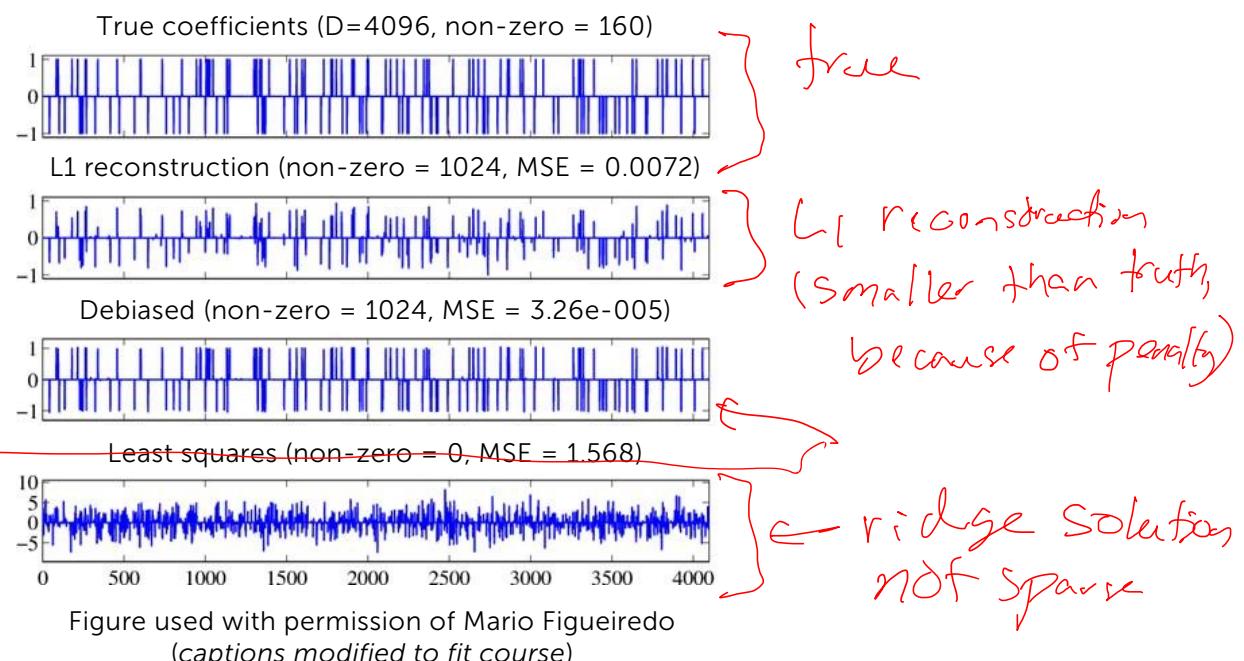
Lasso shrinks coefficients relative to LS solution

→ more bias, less variance

Can reduce bias as follows:

1. Run lasso to select features
2. Run least squares regression with only selected features

"Relevant" features no longer shrunk relative to LS fit of same reduced model



Issues with standard lasso objective

1. With group of highly correlated features, lasso tends to select amongst them arbitrarily
 - Often prefer to select all together
2. Often, empirically ridge has better predictive performance than lasso, but lasso leads to sparser solution

Elastic net aims to address these issues

- hybrid between lasso and ridge regression
- uses L_1 and L_2 penalties

See [Zou & Hastie '05](#) for further discussion

Summary for feature selection and lasso regression

Impact of feature selection and lasso

Lasso has changed machine learning,
statistics, & electrical engineering

But, for feature selection in general, be **careful about interpreting selected features**

- selection only considers features included
- sensitive to correlations between features
- result depends on algorithm used
- there are theoretical guarantees for lasso under certain conditions

What you can do now...

- Describe “all subsets” and greedy variants for feature selection
- Analyze computational costs of these algorithms
- Formulate lasso objective
- Describe what happens to estimated lasso coefficients as tuning parameter λ is varied
- Interpret lasso coefficient path plot
- Contrast ridge and lasso regression
- Implement K-fold cross validation to select lasso tuning parameter λ